**TwinCAT 3软件与MongoDB通讯**

|  |  |
| --- | --- |
|  | 作者：史震宇  职务：技术工程师  邮箱：zy.shi@beckhoff.com.cn  日期： |
| **摘 要：**  TwinCAT3通过TwinCAT Database Server（TF6420）可以实现与数据库通讯，本教程提供了TwinCAT3与MongoDB数据库通讯的方法。  特别注意，使用本样例之前，需要对MongoDB数据库有基本的了解，能够看懂MongoDB的基础指令。 | |
| **附 件：**   |  |  |  | | --- | --- | --- | | 序 号 | 文件名 | 备注 | |  |  |  | |  |  |  | |  |  |  | |  |  |  | | |
| **历史版本：**   |  |  |  | | --- | --- | --- | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | | |
| **免责声明：**  我们已对本文档描述的内容做测试。但是差错在所难免，无法保证绝对正确并完全满足您的使用需求。本文档的内容可能随时更新，如有改动，恕不事先通知，也欢迎您提出改进建议。 | |
| **参考信息：** | |

目 录

[1. 基础知识 3](#_Toc70686493)

[1.1. NoSQL简介 3](#_Toc70686494)

[1.1.1. 什么是NoSQL 3](#_Toc70686495)

[1.1.2. NoSQL的优点与缺点 3](#_Toc70686496)

[1.1.3. 为什么使用NoSQL 3](#_Toc70686497)

[1.1.4. NoSQL数据库的分类 3](#_Toc70686498)

[1.2. MongoDB 3](#_Toc70686499)

[1.2.1. 数据库简介 3](#_Toc70686500)

[1.2.2. MongoDB的特点 4](#_Toc70686501)

[1.2.3. TwinCAT3对MongoDB的支持 4](#_Toc70686502)

[2. 准备工作 4](#_Toc70686503)

[2.1. 软硬件版本 4](#_Toc70686504)

[2.2. Windows平台下MongoDB数据库部署 5](#_Toc70686505)

[2.2.1. 数据库安装 5](#_Toc70686506)

[2.2.2. MongoDB Compass安装与使用 8](#_Toc70686507)

[2.3. Linux平台下MongoDB数据库部署 10](#_Toc70686508)

[2.3.1. Linux服务器直接部署MongoDB 10](#_Toc70686509)

[2.3.2. 通过Docker部署MongoDB 11](#_Toc70686510)

[2.3.3. Mongo-express安装与使用 12](#_Toc70686511)

[2.4. TF6420安装 13](#_Toc70686512)

[2.4.1. 在PC或win7，win10系统的控制器上安装 13](#_Toc70686513)

[2.4.2. 在CE系统的控制器上安装 14](#_Toc70686514)

[3. 操作步骤 14](#_Toc70686515)

[3.1. 在本地PC上配置TF6420数据库连接 14](#_Toc70686516)

[3.2. 在嵌入式控制器上配置TF6420数据库连接 18](#_Toc70686517)

[3.3. 数据库的基本功能测试 21](#_Toc70686518)

[3.3.1. 代码基本格式 21](#_Toc70686519)

[3.3.2. 向数据库插入数据 23](#_Toc70686520)

[3.3.3. 从数据库读取数据 25](#_Toc70686521)

[3.3.4. 更新数据库的数据 27](#_Toc70686522)

[3.3.5. 删除数据库的数据 29](#_Toc70686523)

[3.3.6. 创建或删除数据集 31](#_Toc70686524)

[3.3.7. 数据聚合 32](#_Toc70686525)

[3.4. Json相关指令使用 32](#_Toc70686526)

[3.5. 定期存储数据功能的实现 33](#_Toc70686527)

# 基础知识

## NoSQL简介

### 什么是NoSQL

NoSQL，指的是非关系型的数据库。NoSQL有时也称作Not Only SQL的缩写，是对不同于传统的关系型数据库的数据库管理系统的统称。

NoSQL用于超大规模数据的存储。（例如谷歌或Facebook每天为他们的用户收集万亿比特的数据）。这些类型的数据存储不需要固定的模式，无需多余操作就可以横向扩展。

### NoSQL的优点与缺点

优点：高度可扩展性，分布式计算，低成本，架构灵活，半结构化数据，没有复杂的关系。

缺点：没有标准化，查询功能较为有限。

### 为什么使用NoSQL

今天我们可以通过第三方平台（如：Google,Facebook等）可以很容易的访问和抓取数据。用户的个人信息，社交网络，地理位置，用户生成的数据和用户操作日志已经成倍的增加。我们如果要对这些用户数据进行挖掘，那SQL数据库已经不适合这些应用了, NoSQL 数据库的发展却能很好的处理这些大的数据。

### NoSQL数据库的分类

|  |  |  |
| --- | --- | --- |
| 类型 | 部分代表 | 特点 |
| 列存储 | Hbase Cassandra Hypertable | 列存储的最大特点是方便存储结构化与半结构化数据，方便压缩，对针对某一列或某几列的查询有优势。 |
| 文档存储 | MongoDB CouchDB | 文档存储一般使用类似json的格式存储，存储的内容是文档型的，这样可以对某些字段建立索引，实现类似关系数据库的功能。 |
| Key-Value存储 | Tokyo Cabinet / Tyrant Berkeley DB MemcacheDB Redis | 可以通过key快速查询到其value。一般来说，存储不管value的格式，照单全收。 |
| 图存储 | Neo4J FlockDB | 图形关系的最佳存储。使用传统关系数据库来解决的话性能低下，而且设计使用不方便。 |
| 对象存储 | db4o Versant | 通过类似面向对象语言的语法操作数据库，通过对象的方式存取数据。 |
| Xml数据库 | Berkeley DB XML BaseX | 高效的存储XML数据，并支持XML的内部查询语法，比如XQuery,Xpath。 |

## MongoDB

### 数据库简介

MongoDB官方文档：[Welcome to the MongoDB Documentation — MongoDB Documentation](https://docs.mongodb.com/)。

MongoDB 是由C++语言编写的，是一个基于分布式文件存储的开源数据库系统。在高负载的情况下，支持添加更多的节点，可以保证服务器性能。MongoDB 旨在为WEB应用提供可扩展的高性能数据存储解决方案。MongoDB 将数据存储为一个文档，数据结构由键值(key=>value)对组成。MongoDB 文档类似于 JSON 对象。字段值可以包含其他文档，数组及文档数组。

### MongoDB的特点

MongoDB 是一个面向文档存储的数据库，操作起来比较简单和容易。

可以在MongoDB记录中设置任何属性的索引 (如：FirstName="Sameer",Address="8 Gandhi Road")来实现更快的排序。

可以通过本地或者网络创建数据镜像，这使得MongoDB有更强的扩展性。

如果负载的增加（需要更多的存储空间和更强的处理能力） ，它可以分布在计算机网络中的其他节点上，这就是所谓的分片。

Mongo支持丰富的查询表达式。查询指令使用JSON形式的标记，可轻易查询文档中内嵌的对象及数组。

MongoDb 使用update()命令可以实现替换完成的文档（数据）或者一些指定的数据字段 。

Mongodb中的Map/reduce主要是用来对数据进行批量处理和聚合操作。

Map和Reduce。Map函数调用emit(key,value)遍历集合中所有的记录，将key与value传给Reduce函数进行处理。

Map函数和Reduce函数是使用Javascript编写的，并可以通过db.runCommand或mapreduce命令来执行MapReduce操作。

GridFS是MongoDB中的一个内置功能，可以用于存放大量小文件。

MongoDB允许在服务端执行脚本，可以用Javascript编写某个函数，直接在服务端执行，也可以把函数的定义存储在服务端，下次直接调用即可。

官方支持的语言有：C，C++，C# / .NET，Erlang，Haskell。Java，JavaScript，Lisp，node.JS，Perl，PHP，Python，Ruby，Scala。

### TwinCAT3对MongoDB的支持

TwinCAT3中，需要安装TwinCAT Database Server（TF6420）来实现对MongoDB的支持，安装好function之后，可以使用Tc3\_Database库中的指令对数据库进行增删改查的操作。

# 准备工作

## 软硬件版本

调试PC：TwinCAT3 4024.11，TF6420 V3.3.33.4，mongodb-org-server\_4.4.2，mongodb-compass-1.23.0；

服务器平台：Ubuntu 18.04，mongodb 4.4.4 community

软件安装包下载：

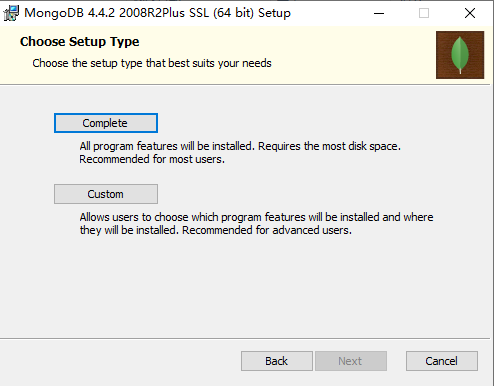
MongoDB 提供了可用于 32 位和 64 位系统的预编译二进制包，可以从MongoDB官网下载安装，MongoDB 源码以及预编译二进制包下载地址：<https://www.mongodb.com/download-center/community>。

TF6420下载地址：<https://www.beckhoff.com/en-en/products/automation/twincat/tfxxxx-twincat-3-functions/tf6xxx-tc3-connectivity/tf6420.html>。

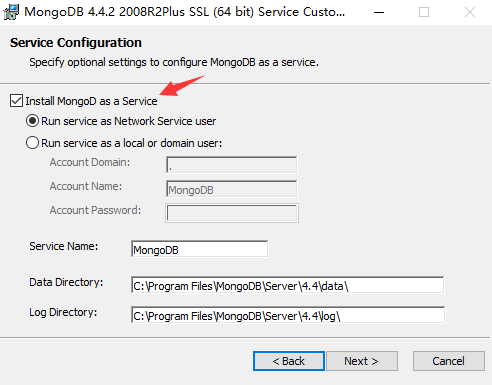
## Windows平台下MongoDB数据库部署

### 数据库安装

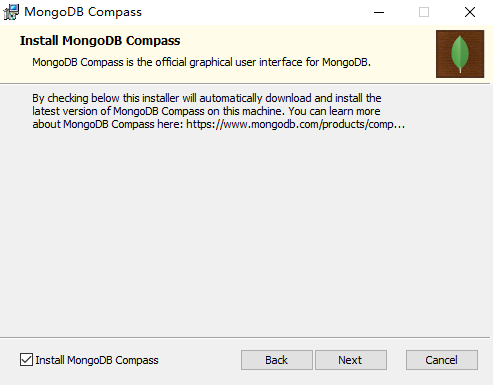
下载官方安装包，执行安装包进行安装即可。安装模式建议选择complete，按照默认配置安装所有组件，安装目录为C:\Program Files\MongoDB\Server\[版本号]，在bin目录下启动mongo.exe即可进入shell操作。；custom安装方式面向相对高阶的用户，可以自定义需要安装的组件，手动选择安装的服务等等，也需要用户手动配置数据库的存储路径和其他配置文件。



确认勾选Install MongoD as a Service，Mongodb将会作为系统服务自动启用。



如果勾选Install MongoDB Compass，在安装的过程中会自动下载MongoDB Compass这个软件并安装，下载会比较费时，建议不要勾选，在MongoDB安装完成之后单独下载安装。



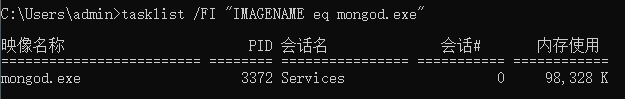
安装完成之后，在C:\Program Files\MongoDB\Server\[版本号]目录下可以看到如下内容：



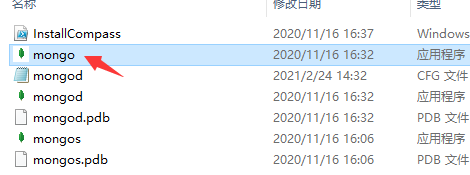
进入终端（CMD），输入如下命令：

tasklist /FI "IMAGENAME eq mongod.exe"

如果返回如下，则说明服务正常启动了：



此时，进入bin目录下执行mongo.exe即可进入mongodb的shell使用数据库命令进行操作了。



首先使用如下命令切换到admin数据集，并且创建root用户：

use admin

db.createUser({user: “<username>”,pwd: “<password>”,roles:[“root”]})

若创建成功显示：

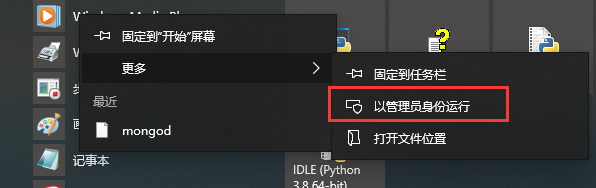


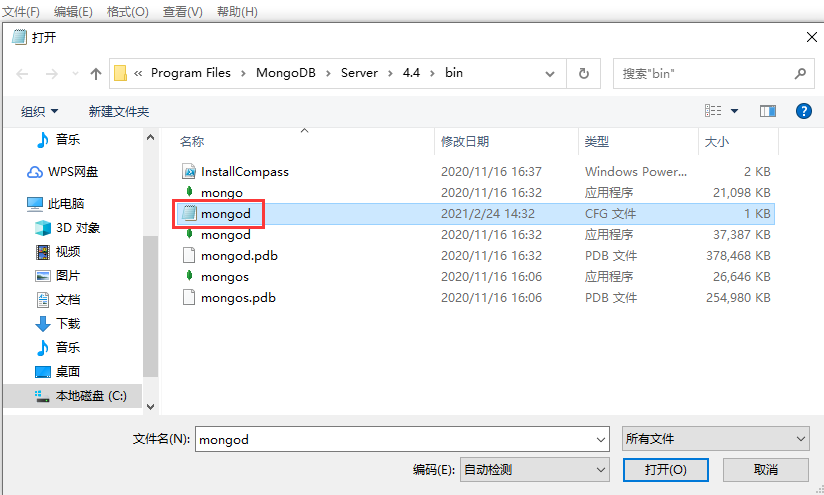
如果要确认用户内容，可以使用“show users”命令。



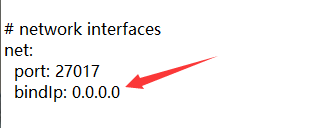
此时数据库已经可以使用，但是仅仅支持本地（localhost）访问，还不支持远程管理，下面需要开启数据库远程访问。

找到记事本工具，右键以管理员权限打开记事本，然后用记事本打开mongodb中bin目录下的mongod.cfg文件。注意，**此处必须以管理员权限打开记事本**，如果没有权限的话，会出现cfg文件无法保存的情况。





将network interfaces下bindIp一栏修改为0.0.0.0：



这样数据库就可以远程访问了，mongodb数据库默认使用的是27017这个端口，如果配置了防火墙等其他安全策略，请保证这个端口的开通。

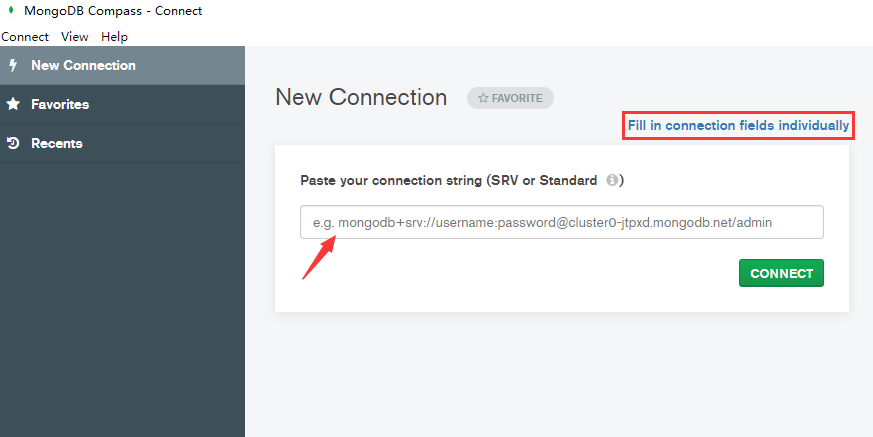
### MongoDB Compass安装与使用

MongoDB Compass 是一个图形界面管理工具，下载地址：<https://www.mongodb.com/download-center/compass>。下载安装完成之后直接使用MongoDBCompass打开。

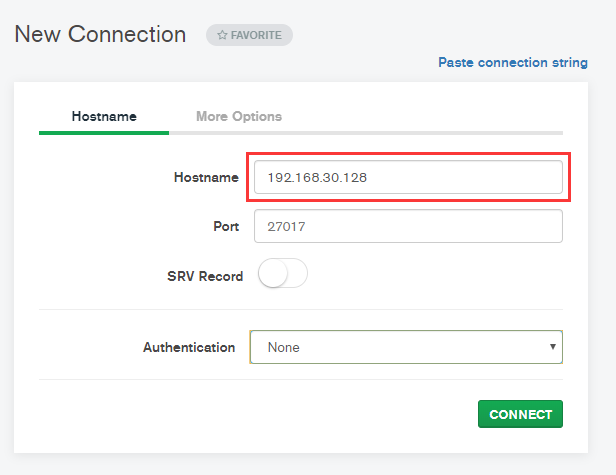
软件打开之后可以新建链接，mongodb通过URL建立连接，语法格式如下：

mongodb://[username:password@]host[:port] [/[database][?options]]

其中username和password即用户名和密码，hosts为服务端的IP地址，port默认为27017，也可以不填，最后为数据库名称以及可选选项，mongodbcompass可以直接输入URL进行连接：

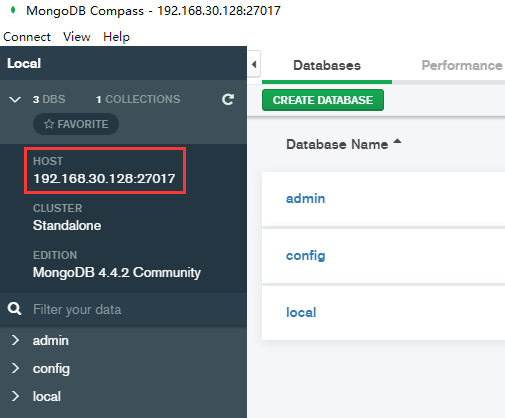


如果不想使用URL，也可以使用右上方的Fill in connection fields individually按钮进入配置界面配置连接参数，软件会自动生成URL进行连接：



Hostname和Port必须填写正确，Authentication可以选择不提供认证，或者是使用用户名密码进行认证，配置完成之后点击connect即可。

连接成功之后显示如下：



可以通过该软件对数据库进行基本的增删改查的操作。

## Linux平台下MongoDB数据库部署

官方安装方法：<https://docs.mongodb.com/manual/administration/install-on-linux/>。

本教程使用Ubuntu 18.04版本进行安装测试，介绍直接部署和通过Docker镜像部署两种方法。

### Linux服务器直接部署MongoDB

本方法参考官方文档：[Install MongoDB Community Edition on Ubuntu — MongoDB Manual](https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/)

打开Ubuntu终端，或者使用SSH远程登录服务器进行操作。

首先导入包管理系统的公钥：

wget -qO - https://www.mongodb.org/static/pgp/server-4.4.asc | sudo apt-key add –

一般来说可以直接导入成功，如果导入失败，提示gnupg未安装，可以使用如下命令进行安装：

sudo apt-get install gnupg

安装完成之后再导入公钥即可。

创建/etc/apt/sources.list.d/mongodb-org-4.4.list文件：

echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.4 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.4.list

重载包管理器数据库：

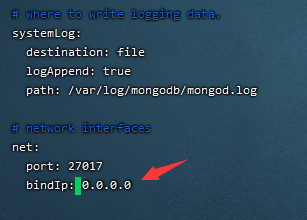
sudo apt-get update

刷新包管理数据库之后，即可直接使用apt-get安装数据库：

sudo apt-get install -y mongodb-org

安装完成之后，即可使用mongodb数据库，不过和windows平台相同，需要修改一下mongodb的配置文件，使用如下命令修改配置文件：

vi /etc/mongod.conf



将net栏目下bindIp一项修改为0.0.0.0，这样数据库就可以允许外网访问了。

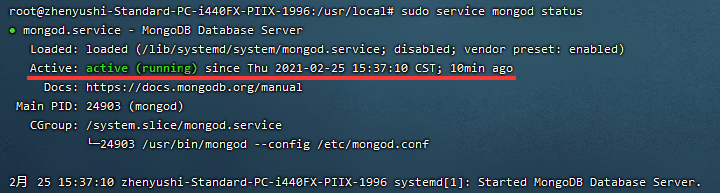
最后，使用命令启动mongod服务：

sudo service mongod start

如果需要查看服务状态，可以使用如下命令：

sudo service mongod status

如果返回如下，说明服务正常运行中：



此时，也可以用mongo-compass对该数据库进行连接，更多说明请参考官方文档。

### 通过Docker部署MongoDB

Docker是一个开源的应用容器引擎，基于Go语言并遵从Apache2.0协议开源。

Docker可以让开发者打包他们的应用以及依赖包到一个轻量级、可移植的容器中，然后发布到任何流行的Linux机器上，也可以实现虚拟化。简单来说，Docker容器可以被理解为一个独立且隔离的小型虚拟化系统，每一个容器对应一个单独的服务来执行。容器是完全使用沙箱机制，相互之间不会有任何接口（类似iPhone的app）,更重要的是容器性能开销极低，基本上没有性能损耗。

通过Docker部署服务的好处在于，每个服务的环境是相对独立的，互相不会发生干涉；且发布者在发布镜像的时候，在镜像中将相应的环境配置等内容安装完成，可以避免用户进行繁琐的设置，使用者只要拉取镜像，配置最基本的端口映射，磁盘映射以及少许环境变量，即可使用。

只要在系统上部署好Docker引擎，就可以无差别的使用Docker来实现各式各样的服务。MongoDB也支持以Docker这种形式来部署。

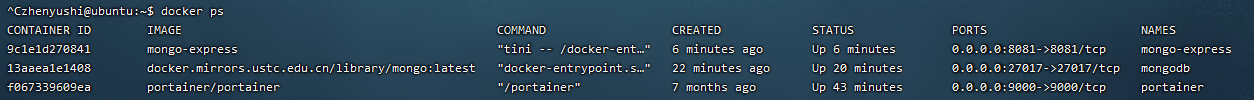


Ubuntu平台部署Docker引擎的命令如下：

curl -fsSL https://get.docker.com | bash -s docker --mirror Aliyun

其他平台部署Docker引擎的方法可以自行查询。

安装成功之后，就可以使用docker ps命令来查看容器了（下图为有容器正在运行时的情况）。



如果是新安装的环境，则不会有正在运行的容器，docker ps显示内容为空。

下面可以开始部署mongodb的官方镜像，mongodb官方镜像的dockerhub链接如下：

<https://hub.docker.com/_/mongo>

直接输入以下命令：sudo docker run -it --name mongodb -v [database]:/data/db -p 27017:27017 --restart unless-stopped mongo:latest

其中[database]填写数据库文件保存的路径，按需配置即可。

系统会自动拉取mongodb的镜像并且运行镜像，整个过程是完全自动的。部署完成之后，直接连接服务器的27017端口即可使用mongodb数据库。

### Mongo-express安装与使用

通过Docker可以配置一个简单方便的mongodb网页端管理工具——mongo-express，该容器的dockerhub链接：

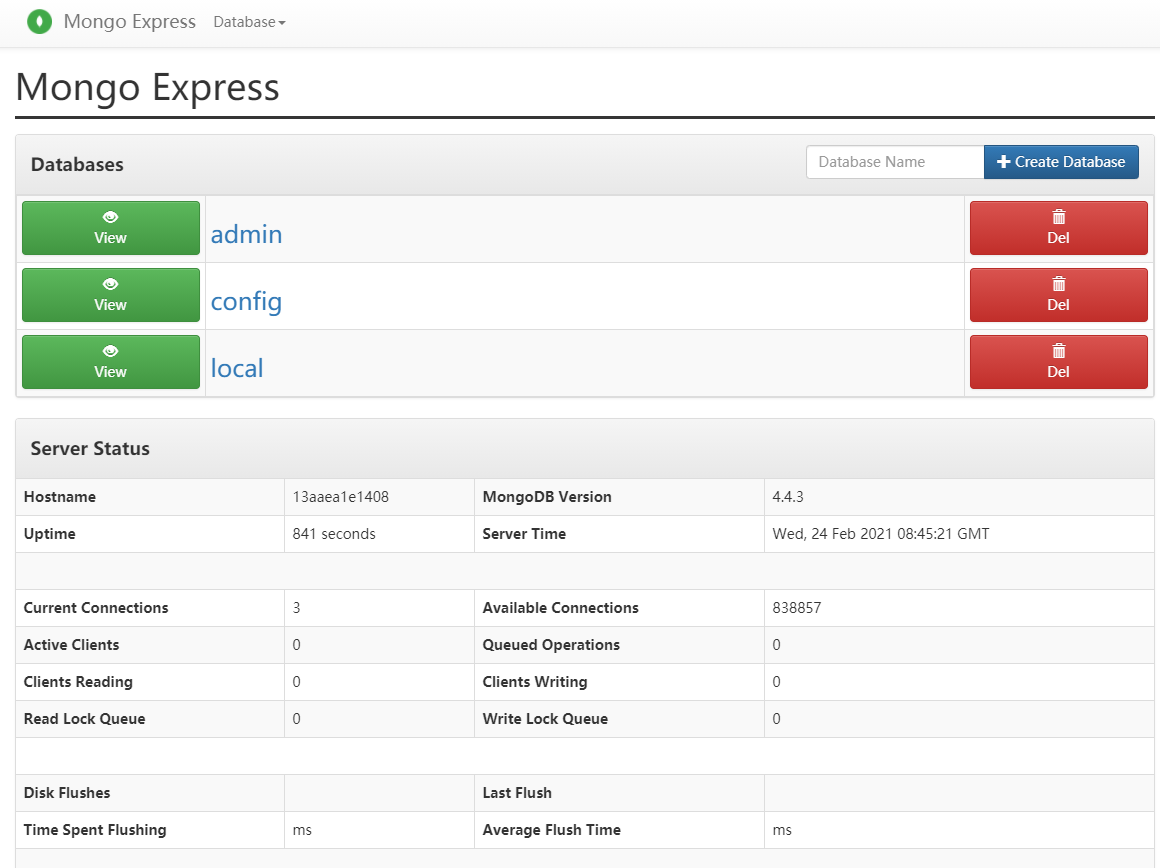
<https://hub.docker.com/_/mongo-express>

部署命令如下：

sudo docker run -it --name mongo-express -p 8081:8081 -e ME\_CONFIG\_MONGODB\_SERVER=[IP] -e ME\_CONFIG\_MONGODB\_PORT=[PORT] --restart unless-stopped mongo-express:latest

其中[IP]填写数据库服务器的IP地址，[PORT]填写mongodb数据库的端口。

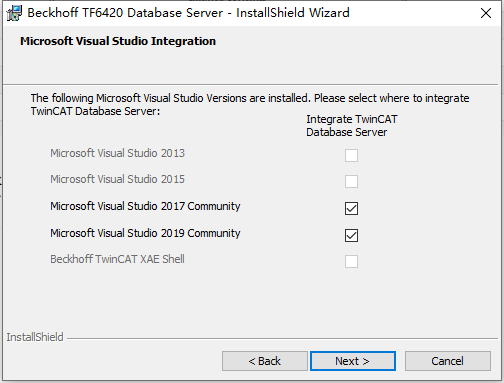
该过程也是完全自动的，部署完成之后访问服务器的IP+8081端口即可管理数据库的内容：



## TF6420安装

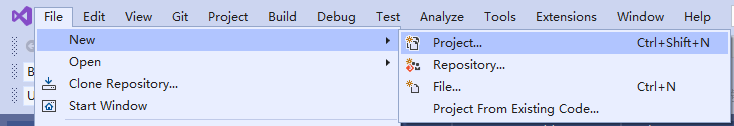
### 在PC或win7，win10系统的控制器上安装

下载好安装包之后，以管理员权限执行安装包进行安装。注意在安装的时候，要选择嵌入到当前的VS环境中：

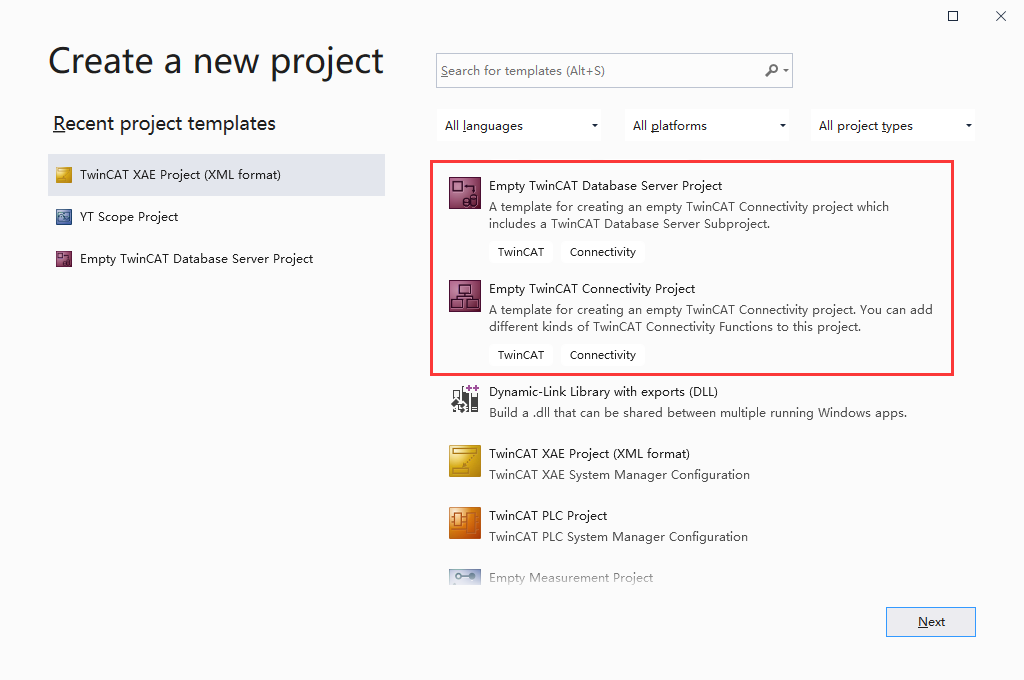


其余选项全部按照默认，最后选择install即可安装。

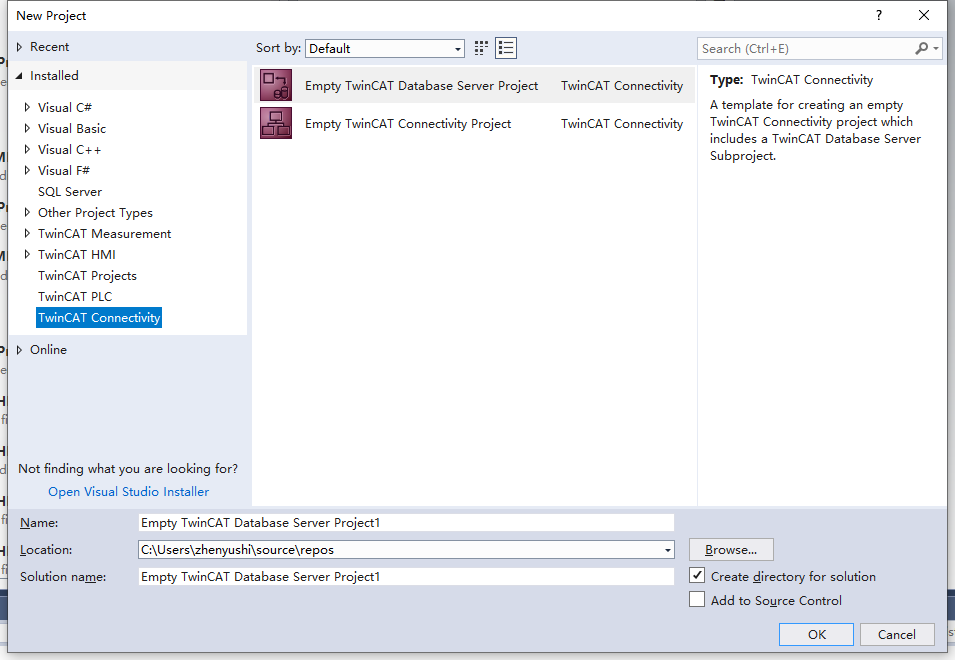
在编程电脑上安装完成之后，打开VS编程环境，选择File-New-Project新建项目：



可以看到已经可以新建TwinCAT Database Server Project，说明安装成功。



如果使用的是VS2017或者TwinCAT XAE shell，则如下图：



### 在CE系统的控制器上安装

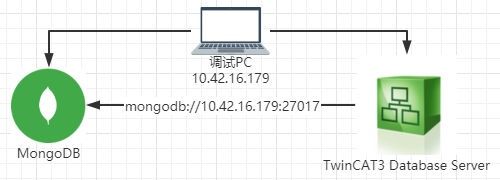
CE系统下安装Function的方法参考如下连接：

<https://tr.beckhoff.com.cn/pluginfile.php/22735/mod_folder/content/0/Windows%20CE%E7%B3%BB%E7%BB%9F%E4%B8%8B%20Supplement%E7%9A%84%E5%AE%89%E8%A3%85%E6%96%B9%E6%B3%95V2.docx?forcedownload=1>

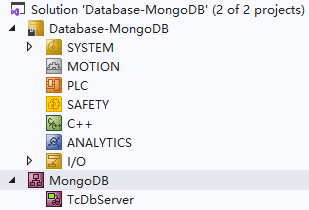
# 操作步骤

## 在本地PC上配置TF6420数据库连接

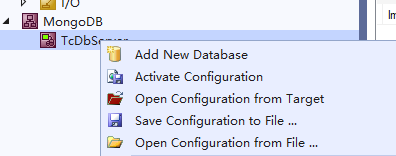
如果没有控制器，本样例支持在本地PC进行模拟运行和调试，此时，PLC程序和数据库都在本地PC上运行，PC上需要安装TF6420。拓扑如下：



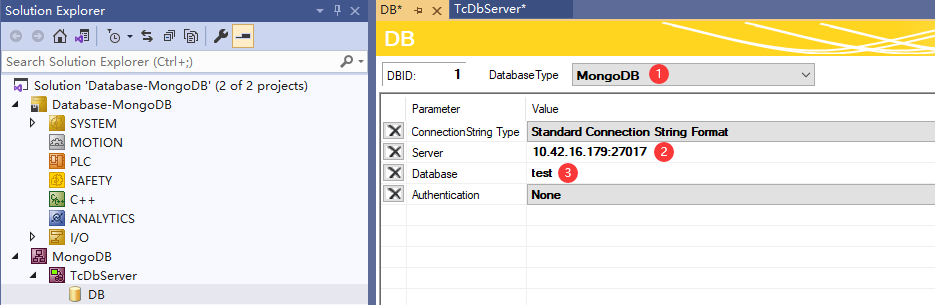
首先新建一个TwinCAT Project，命名：Datebase-MongoDB;以及Database Project, 命名：MongoDB：



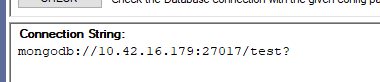
右键TcDbServer，选择Add New Database添加数据库：



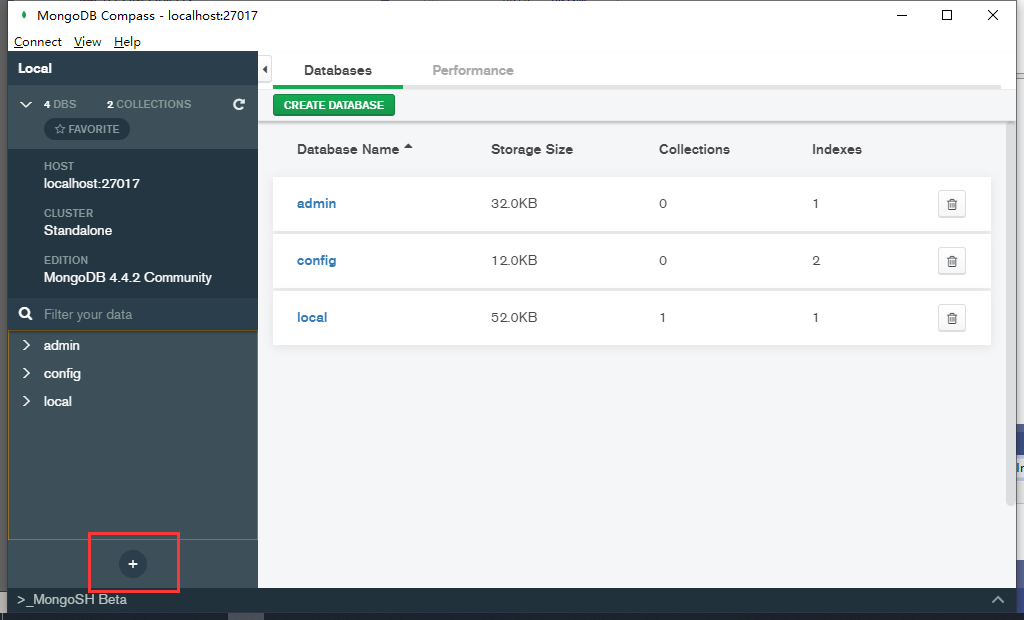
配置数据库连接，数据路类型选择MongoDB，Server一栏填写数据库的IP+端口，这里由于数据库部署在本地电脑，IP填写localhost或电脑的IP地址即可，Database一栏输入数据库名称，本次实验不配置验证用的账户和密码，在实际应用场景中请务必配置数据库账户或其他验证方式，账户信息可以在Authentication一栏选择用户名密码并填写。



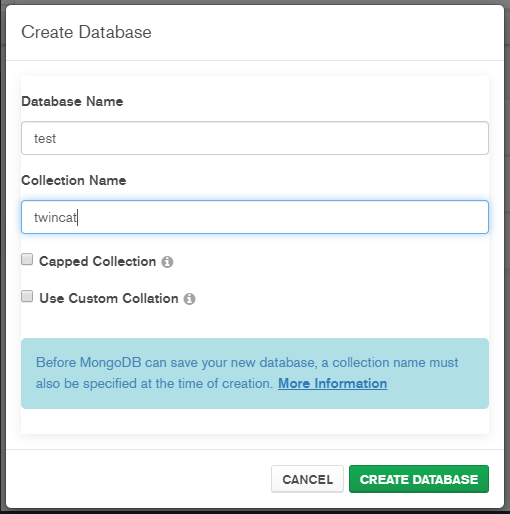
配置完成之后可以看到TwinCAT会自动生成连接用的命令：



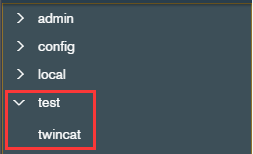
在MongoDB一侧，需要设置给TwinCAT使用的数据库，打开MongoDB Compass，连接到目标数据库，选择左下角的+号，新建一个Database：



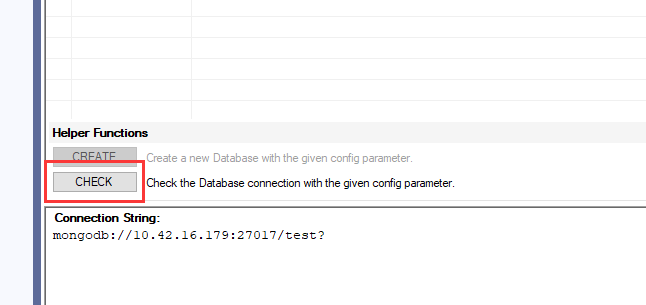
Database名称选择test，数据集名称填写twincat：



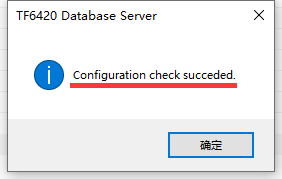
数据库已经建立成功：



可以使用左下方的CHECK按钮来测试数据库的连接状态：



如果出现Configuration check succeed，则说明配置无误，数据库连接正常。

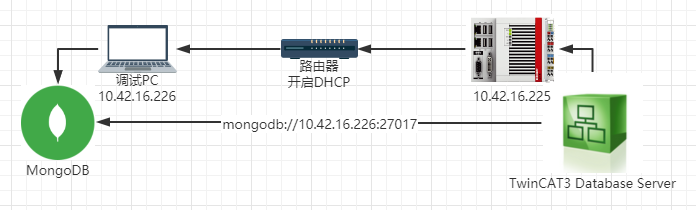


确认无误之后激活database的配置：



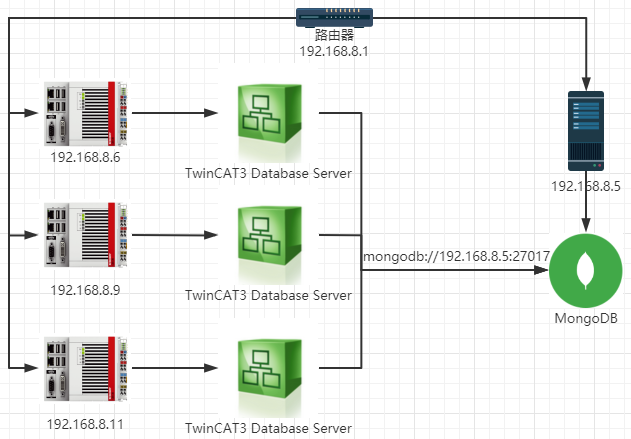
## 在嵌入式控制器上配置TF6420数据库连接

本样例测试用的拓扑图如下：

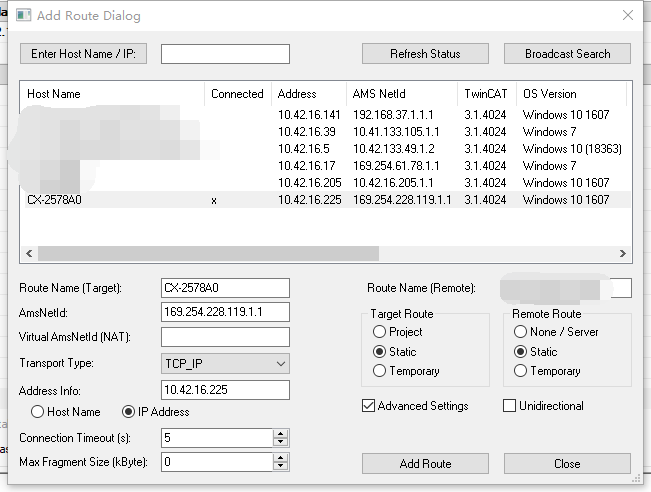


调试PC和嵌入式控制器接入同一个路由器，路由器开启DHCP功能为PC和控制器分配IP，这样控制器和PC处于同一个局域网中，互相能够访问，mongodb数据库部署在调试PC上；注意，如果采用PC网线直连控制器这种方式，则控制器访问调试PC上的数据库时会报超时错误，造成无法连接数据库的情况。

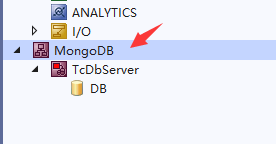
如果有多台控制器，也支持多台控制器连接到同一个数据库，如下图所示：



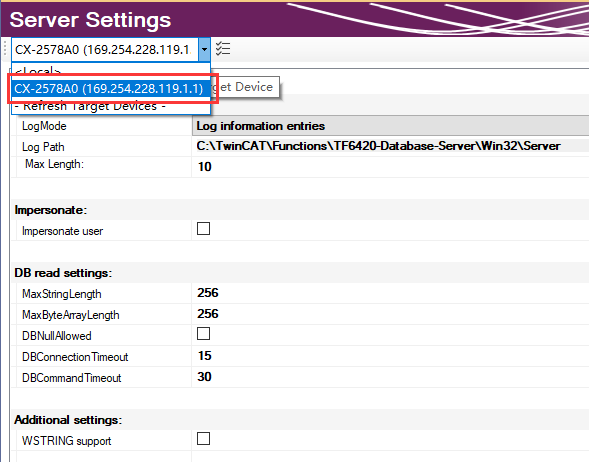
在嵌入式控制器上安装好TF6420之后，通过电脑连接控制器：



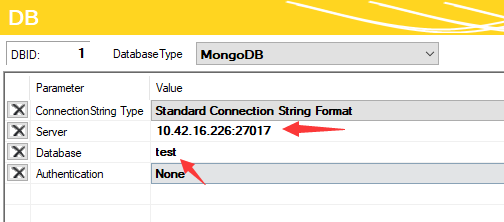
添加database项目：



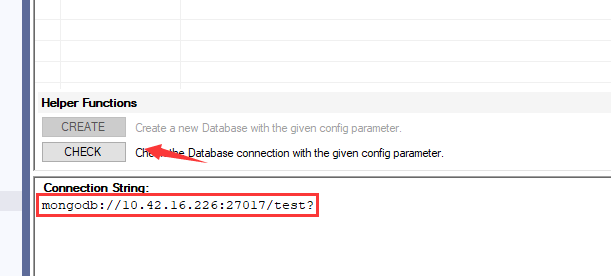
TcDbServer需要选择目标控制器：



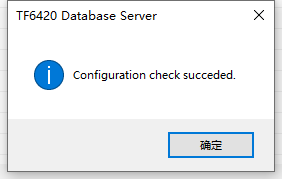
数据库类型选择MongoDB，Server填写数据库所在的服务器以及端口，由于本次实验数据库在调试PC上，直接填写调试电脑的IP即可，如果有独立数据服务器，需要按照实际服务器的情况来填写，Database填写目标数据库名称：



确认连接命令正确，然后点击check检查连接状态：



出现configuration check succeded，说明和数据库连接正常。



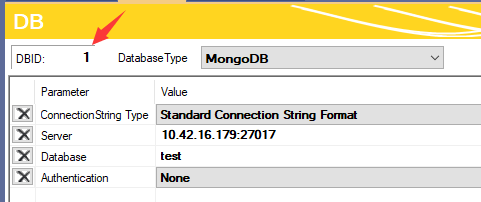
确认连接成功之后激活database的配置。



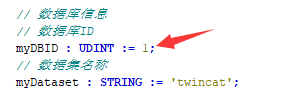
## 数据库的基本功能测试

数据库连接成功之后，就可以通过PLC程序对数据库实现增删改查的操作了，本节会提供相应操作的样例代码。

开始调试程序之前，需要在Database配置界面记录DBID这个参数，然后将这个值填写到PLC程序中：



测试用的数据集为twincat数据集。



完成之后就可以login程序开始调试了。

### 代码基本格式

在TwinCAT中，配置数据库通讯的命令分为几部分，这里以向数据库写入数据为例讲解。

MongoDB数据库写入参数的命令如下：

db.inventory.insertOne(

{ "item" : "canvas",

"qty" : 100,

"tags" : ["cotton"],

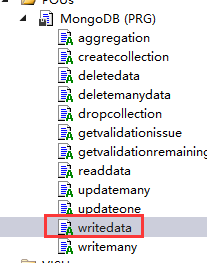
"size" : { "h" : 28, "w" : 35.5, "uom" : "cm" }

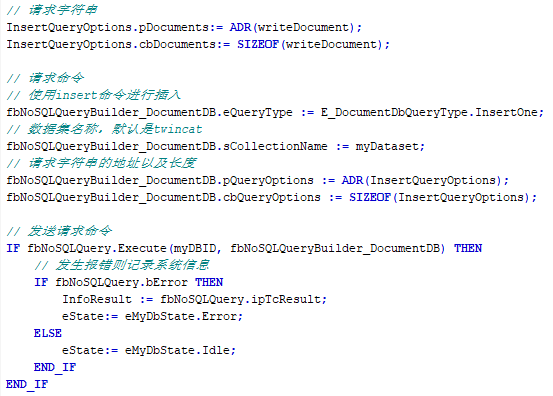
}

)

该命令分为两部分，第一部分“db.inventory.insertOne(<data>)”为数据库操作命令，即插入一条数据；后面括号内的参数（<data>部分）为Json格式文档，即insertOne指令的后续参数，对于插入数据指令而言，就是需要插入到数据库中的文档内容。

在TwinCAT3中搭建该命令的程序如下：





将命令的参数（即要写入的字符串）注入结构体T\_QueryOptionDocumentDB\_Insert（变量InsertQueryOptions）；然后使用FB\_NoSQLQueryBuilder\_DocumentDB（功能块fbNoSQLQueryBuilder\_DocumentDB）搭建命令，其中eQueryType选择InsertOne，即插入参数的函数，sCollectionName参数填写数据集的名称，QueryOptions填写结构体T\_QueryOptionDocumentDB\_Insert的地址以及长度。

由于该命令没有返回值，因此使用FB\_NoSQLQueryEvt的Execute（）函数进行触发，触发的参数需要填写DBID以及FB\_NoSQLQueryBuilder\_DocumentDB。这样就成功发送了一条插入参数的命令。

如果是查找参数，则该命令会返回数据库的查询结果，此时需要使用FB\_NoSQLQueryEvt的ExecuteDataReturn（）函数进行触发，触发参数需要填写DBID，FB\_NoSQLQueryBuilder\_DocumentDB，以及FB\_NoSQLResultEvt，然后使用FB\_NoSQLResultEvt的ReadAsString来处理返回值参数。

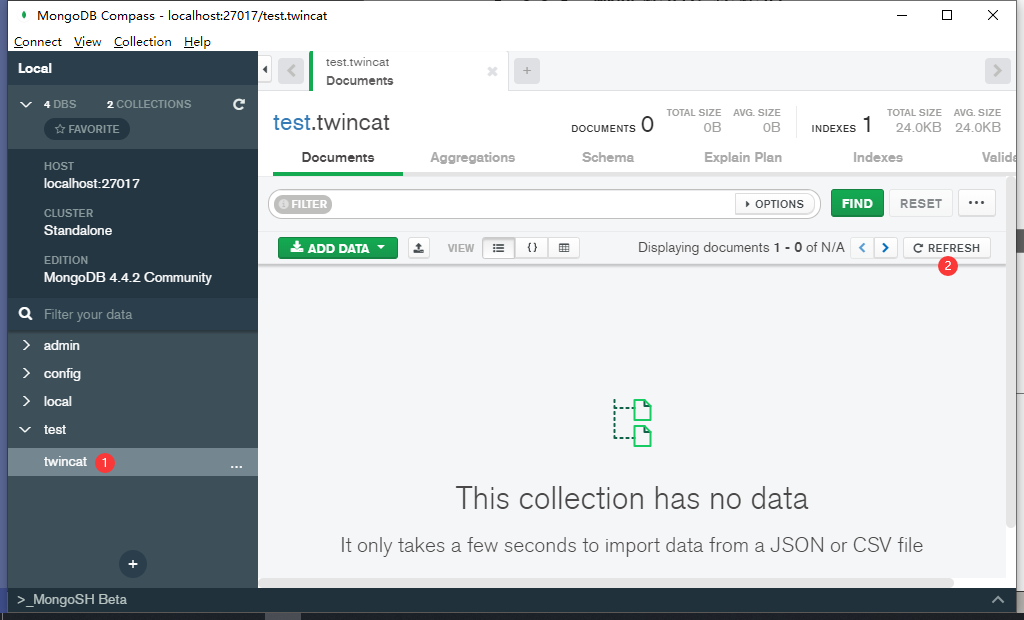


### 向数据库插入数据

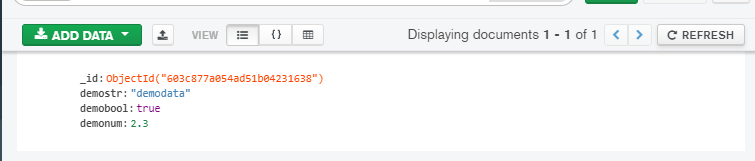
将eState置为write来调用writedata方法：



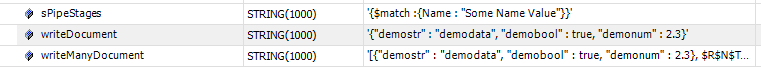
该方法会将writeDocument变量中的Json格式数据传入MongoDB，操作完成之后使用MongoDB Compass连接数据库，进入对应的数据集，点击REFRESH进行刷新：



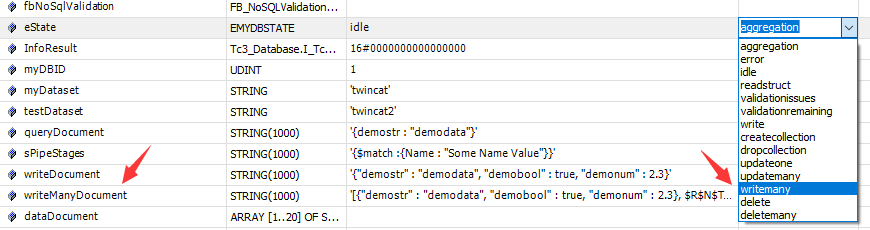
可以看到参数有所更新：



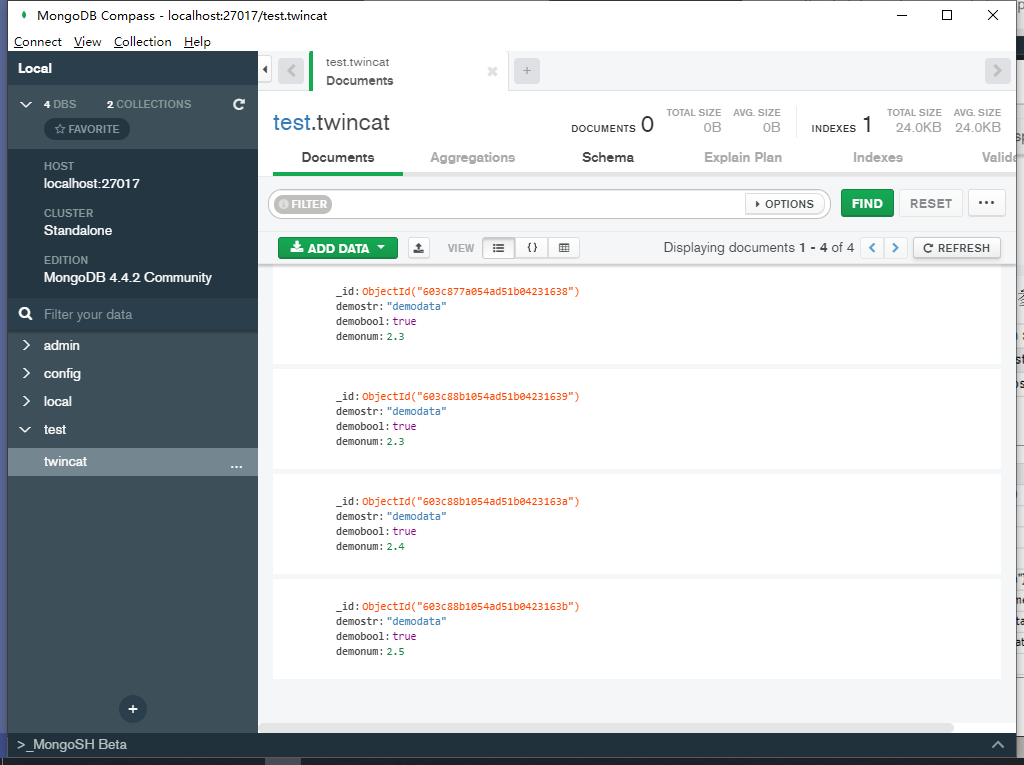
数据库会自动为其中的Json对象添加ObjectId，其余参数部分与writeDocument变量的值一致：



此外，也可以调用多次插入的方法：

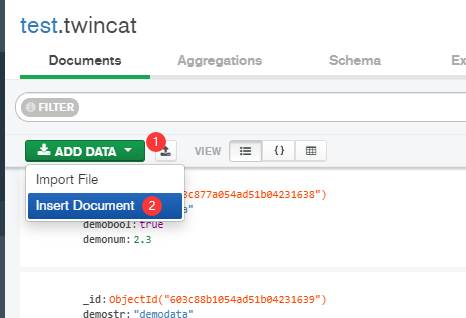


在数据库中一次性插入了多组参数；

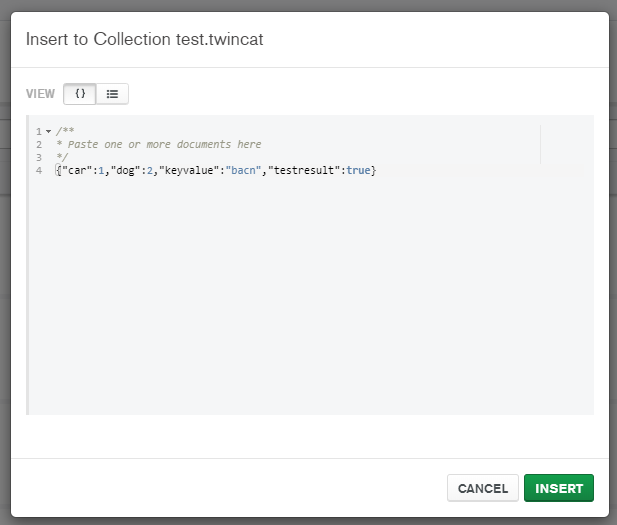


### 从数据库读取数据

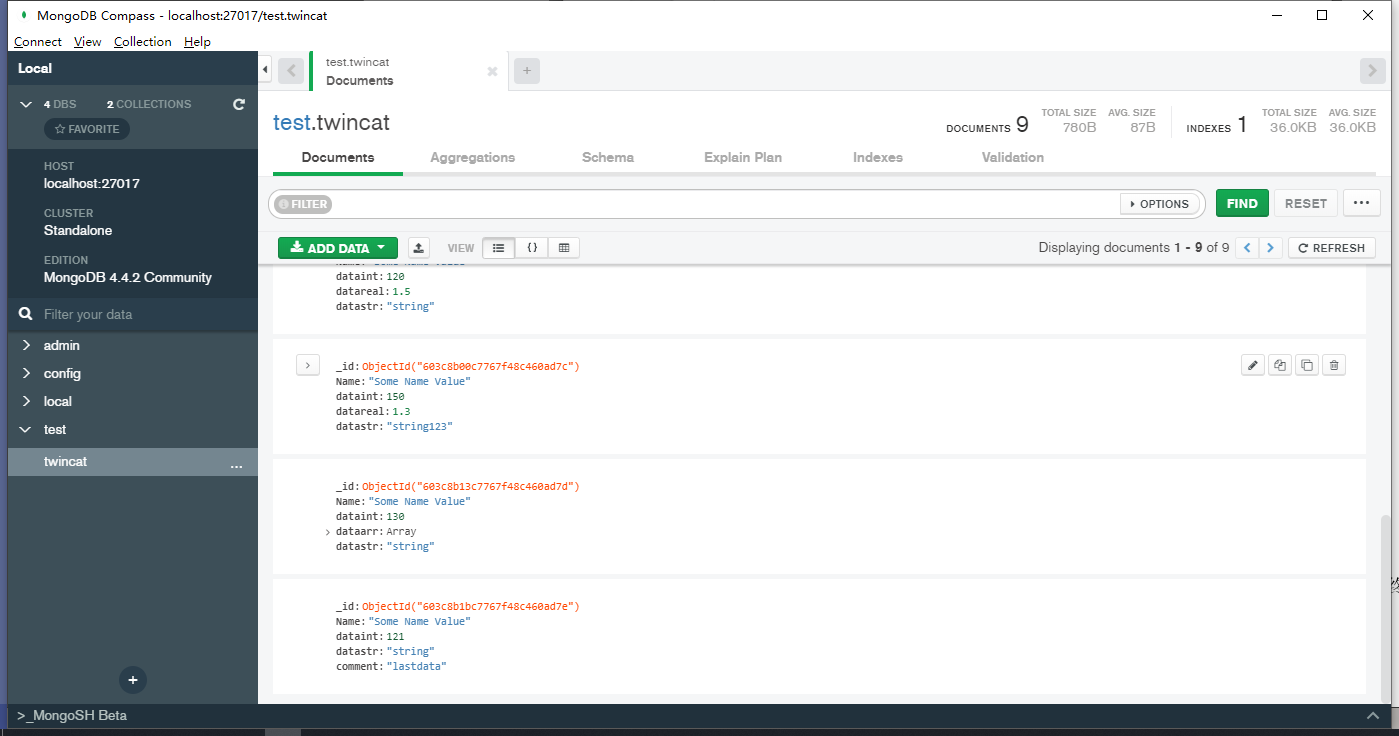
在进行读取测试之前，建议先手动向数据库中写入一些参数，选择Add Data-Insert Document就可以手动插入数据了：



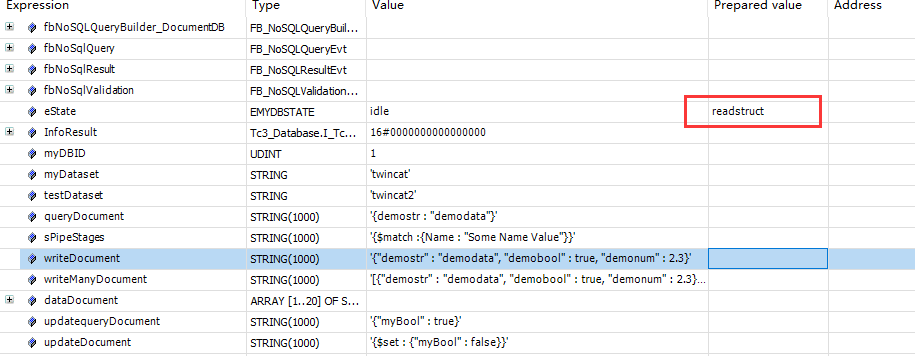
手动插入一些Json数据：



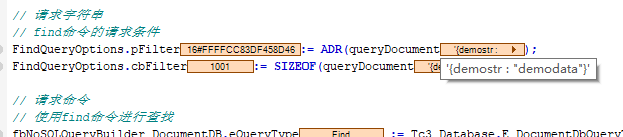
通过以上方法手动插入多组数据，可以将提供的json文件中的数据稍作修改并加入数据库，最终结果如图：

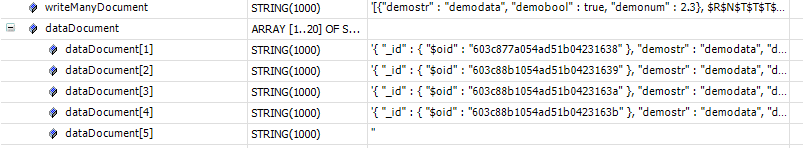


切换到read方法：



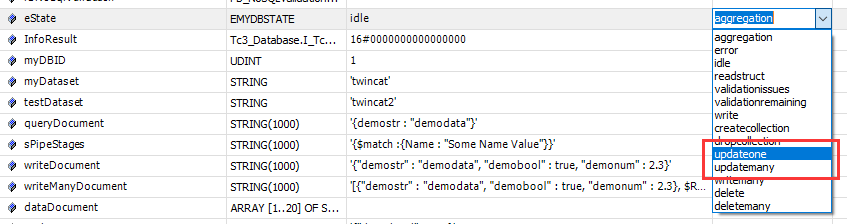
由于在find命令的请求参数中，设置了条件，即demostr一项的值为demodata，因此数据库返回了四组数据，它们被存储在dataDocument中：



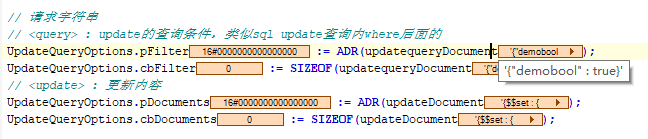


### 更新数据库的数据

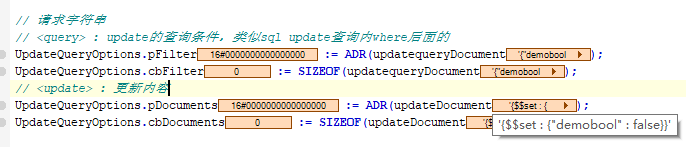
使用updateone方法以及updatemany方法可以更新 单组/多组参数：



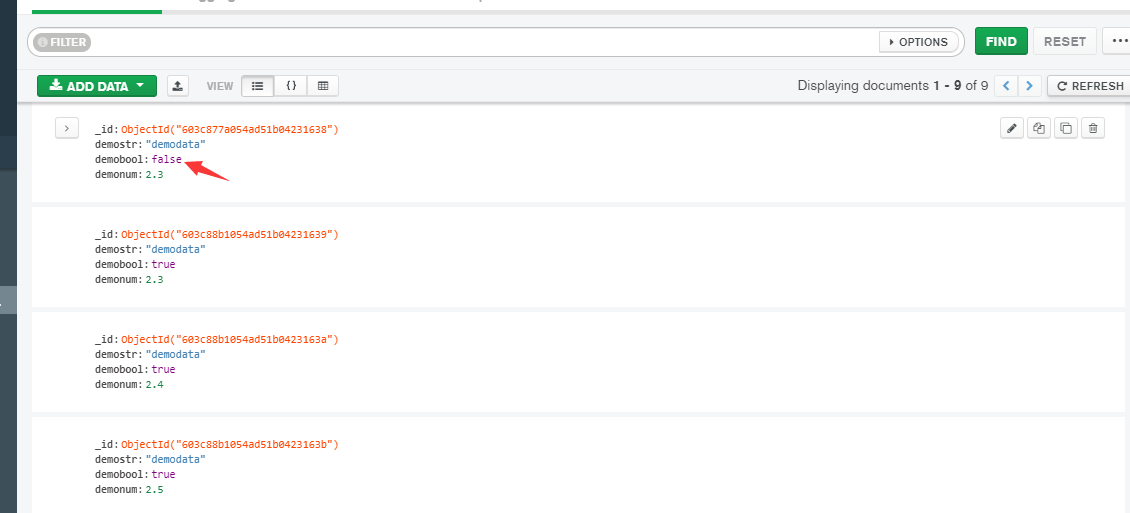
在update方法中，配置了查找条件如下：



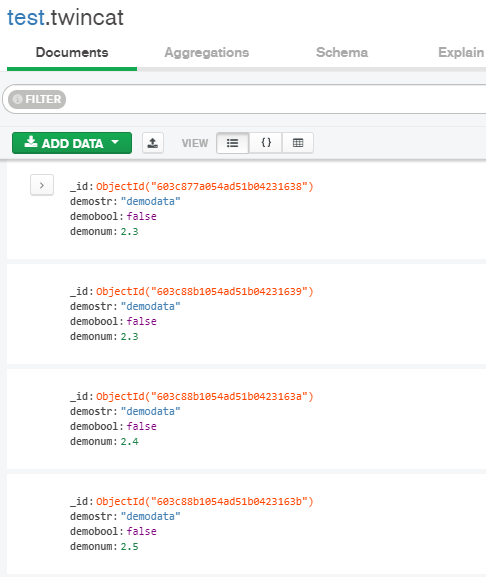
更新内容如下：



调用updateone方法的效果，可以看到第一组参数中的demobool参数被更新为false：



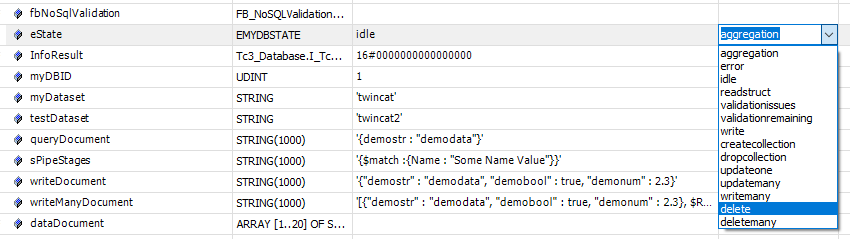
如果使用updatemany方法，效果如图：



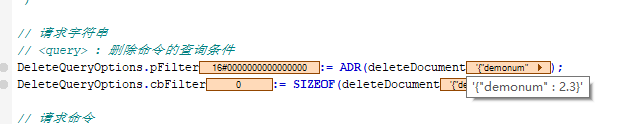
多组参数都得到了更新。

### 删除数据库的数据

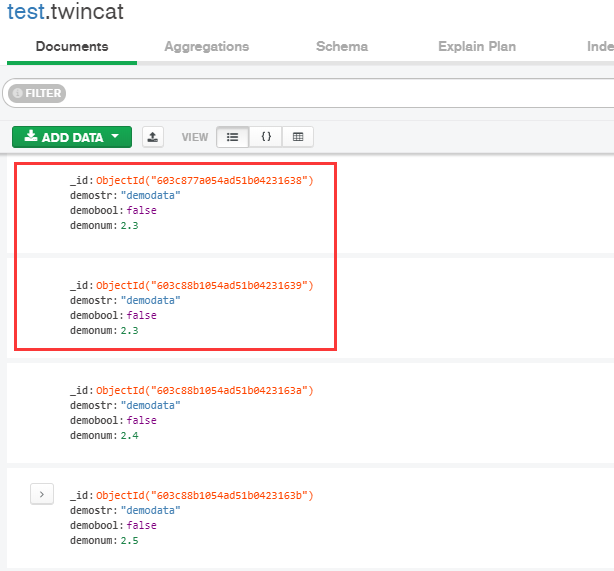
使用deletedata以及deletemanydata方法，可以删除单个或者多个数据：



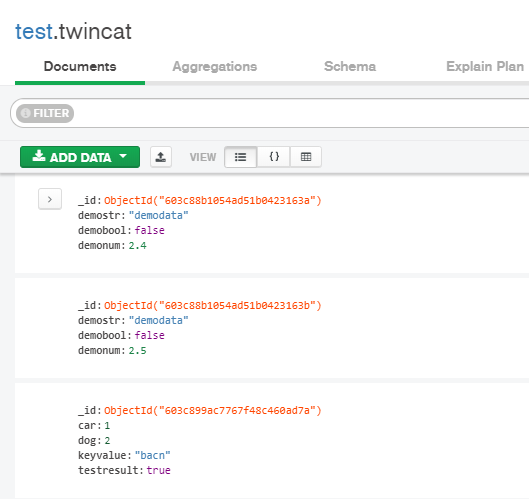
其中删除的条件为demonum一项的值为2.3：



可以看到数据库中有两项符合的数据：



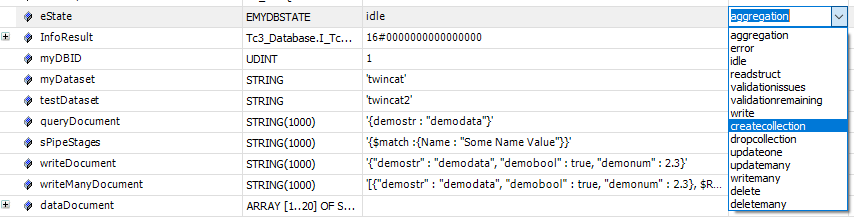
执行deletemanydata方法后：



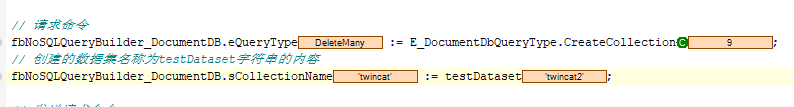
可以看到两组数据已经被删除。

### 创建或删除数据集

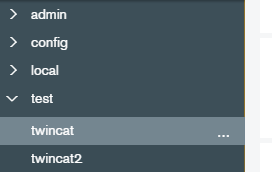
通过createcollection以及dropcollection可以创建/删除数据集：



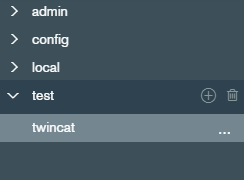
创建一个名为twincat2的数据集：



数据集创建成功：



使用drop命令将其删除：

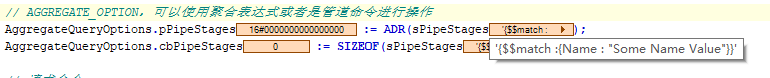


数据集已经被清除。

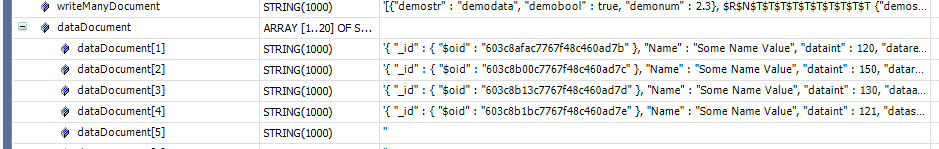
### 数据聚合

在MongoDB中聚合主要用于处理数据，比如说统计平均值，求和等操作，然后返回计算的结果，类似于SQL语句中的count命令。也可以搭配多种管道命令来使用。

这里使用match命令，可以实现类似于find查找的效果：



返回值被填写在dataDocument这个数组当中：



## Json相关指令使用

MongoDB使用类似json的格式存储，因此在读写MongoDB的时候需要将数据转换为Json格式，将Json字符串解析为TwinCAT能识别的数据类型，倍福虚拟学院中提供了Json数据操作的教程，本文也将使用该教程中提供的Json处理指令，参考链接：

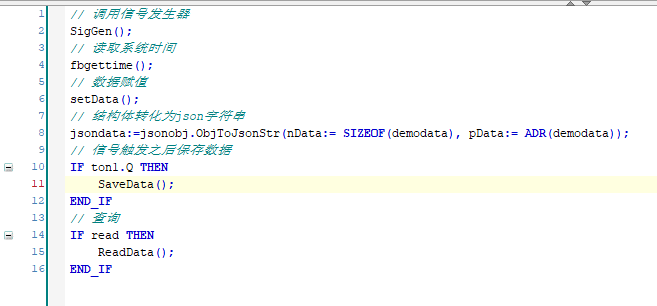
<https://tr.beckhoff.com.cn/mod/folder/view.php?id=2220>



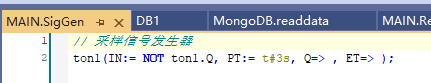
## 定期存储数据功能的实现

打开样例程序2，该程序实现了定期存储数据的功能，在使用该程序之前，需要建立一个名为datasave的数据库以及twincat数据集并测试连接。

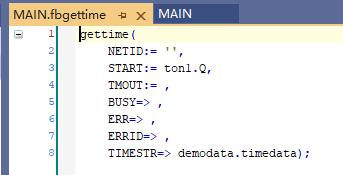
本样例模拟了一个信号发生器，并且对于信号发生器输出的值做定期的采样和记录，并且对值进行一定的判断（判断输出过高或者正常），需要记录的数据为，信号发生器的输出值，记录时间点，对信号值的判断/描述。



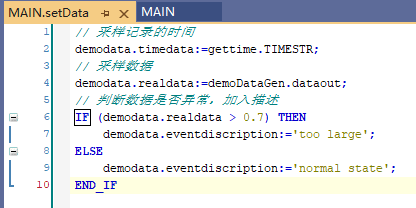
使用TON做脉冲信号发生器，生成3s一次的脉冲信号，用于触发采样动作：



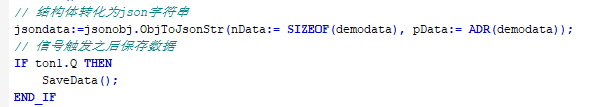
使用NT\_gettime功能块读取系统时间：



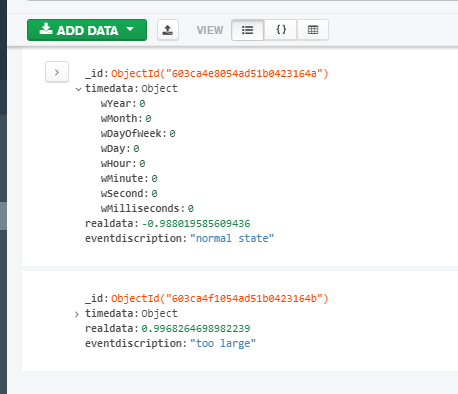
将数据写入指定的结构体对象，并且根据信号值判断信号状态，信号大于0.7则认为信号大于正常值。



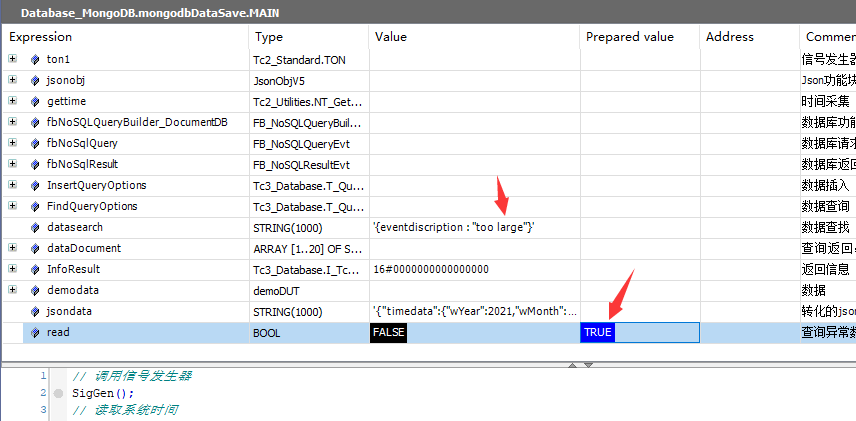
使用Json功能块将结构体转化为json对象，并且写入数据库。

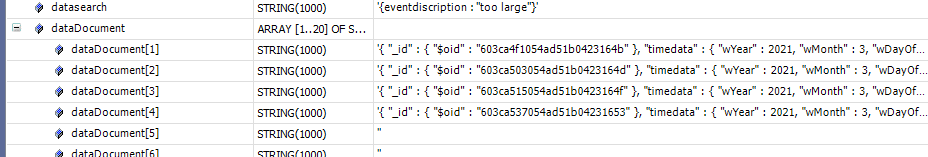


程序运行起来之后，每隔3秒会采样并存储一次数据，数据结构的内容包括采样时间，数据值，以及数据描述：

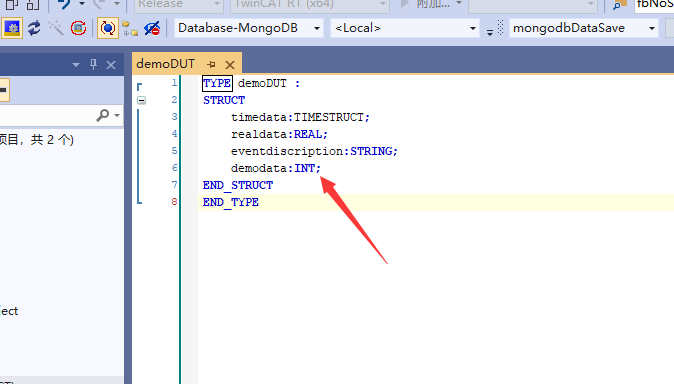


触发read变量可以从数据库中搜索数据，自动将值过大的数据都回传到twincat。





采用这种方式可以将数据定期采样并存放到目标数据库中，且mongodb支持的高级语言非常丰富，官方支持的语言有C，C++，Python等十余种，可以方便地与各种高级语言交互。且mongodb使用json文档作为存储形式，可以很方便地进行数据结构的变动，比如在结构体中增加一个数据，然后**只需要配置该数据相关的采样程序**，重新运行项目，该数据就会自动存储到数据库中，且**不影响原有数据库中的数据结构**。



**上海（ 中国区总部）**

中国上海市静安区汶水路 299 弄 9号（市北智汇园）

电话: 021-66312666

**北京分公司**

北京市西城区新街口北大街 3 号新街高和大厦 407 室

电话: 010-82200036 邮箱: beijing@beckhoff.com.cn

**广州分公司**

广州市天河区珠江新城珠江东路16号高德置地G2603 室

电话: 020-38010300/1/2 邮箱: guangzhou@beckhoff.com.cn

**成都分公司**

成都市锦江区东御街18号 百扬大厦2305 房

电话: 028-86202581 邮箱: chengdu@beckhoff.com.cn

|  |  |
| --- | --- |
| 请用微信扫描二维码  通过公众号与技术支持交流 | 倍福官方网站：  https://www.beckhoff.com.cn  在线帮助系统：  https://infosys.beckhoff.com/index\_en.htm |
| 倍福虚拟学院：  https://tr.beckhoff.com.cn/ |
| 招贤纳士：job@beckhoff.com.cn  技术支持：support@beckhoff.com.cn  产品维修：service@beckhoff.com.cn  方案咨询：sales@beckhoff.com.cn |
|  |